

# SMALL SOFTWARE SYSTEM

# RSM SYSTEM MONITORS

Based on a machine language monitor program widely used in S-100 computers, RSM monitors allow you to interact directly with your TRS-80 at the machine language level. You may examine your Basic ROM's, test your RAM, enter and execute machine language programs, read and write machine language tapes, and much more!

There are currently 4 RSM monitors, RSM-1, RSM-1S, RSM-2, and RSM-2D. RSM-1/1S are for 4K computers, RSM-2 is for 16K tape computers, and RSM-2D is for computers with disk systems. RSM-1/1S/2 will operate in either LEVEL-I or LEVEL-II computers. RSM-2D requires LEVEL-II.

Functionally, the four monitors are nearly identical. The advanced versions have more commands. RSM-2/2D contain their own video drivers, keyboard scanners, and cassette drivers, thus they do not access BASIC routines. RSM-1/1S interact with BASIC in order to conserve memory. The following table is a memory map of the RSM monitors (note that there are 3 versions of RSM-2D on each diskette):

MONITOR	START ADDRESS	END ADDRESS	USER ADDRESS
RSM-1	4200	4880	4880
RSM-1S	4200	4FB0	4FB0
RSM-2	6C00	7EFF	7F80
RSM-2D (16K)	6C00	7EFF	7F80
RSM-2D (32K)	AC00	BEFF	BF80
RSM-2D (48K)	EC00	FEFF	FF80

See the USER command for significance of the USER ADDRESS.

## LOADING INSTRUCTIONS

**RSM-1/1S:** For LEVEL-I users, load the program using the TRS-80 CLOAD command. The tape will load and execute automatically. For LEVEL-II, use the SYSTEM command, which is documented in your BASIC manual. The file name is RSM and the start address is 16896. After loading, type / <ENTER> and RSM will execute.

**RSM-2:** This version is furnished in our 2-level tape format, which will load into either LEVEL-I or LEVEL-II machines. A short tape loading routine is recorded at the LEVEL-I rate of 250 baud, followed by the main program at 500 baud. Load into LEVEL-I using CLOAD. When the tape starts to load, two stars will appear in the upper left of the screen. After about 5 seconds, the screen will clear, indicating the loader is running. 5 seconds later when program loading begins, two stars will appear in the upper right corner, with the right one blinking every 4 seconds. Execution is automatic after loading is complete. If you have LEVEL-II, use the SYSTEM command (see your BASIC manual). The name is RSM and the start address is 27648. Your LEVEL-II computer will ignore the LEVEL-I loader at the beginning of the tape.

If your cassette fails to load, refer to "IN CASE OF TAPE DIFFICULTY" near the end of this documentation.

\* \* \* WARNING! \* \* \*

Do NOT leave your diskettes in the disk drive when you turn off your TRS-80. We have found (the hard way) that even write protected disks can be damaged!

## RSM-2D BACKUP DISKS

We strongly recommend that you do not remove the write-protect tab from the RSM-2D disk. A backup copy, using the following procedure, should be made and the original disk stored in a safe place. See page 7 of your manual for L and \$ command explanations. Please note that back-up copies of RSM-2D are authorized for your own use only!

- 16K SYSTEMS:
- 1) Format a diskette with your TRSDOS FORMAT program.
  - 2) Load RSM-2D, version 1 (16K version).
  - 3) Type: L 0 4000 30 <ENTER> to load 30 sectors into memory.
  - 4) Insert your formatted diskette.
  - 5) Type: \$ 0 4000 30 <ENTER> to save 30 sectors to your backup disk.
  - 6) Replace the RSM-2D diskette.
  - 7) Type: L 30 4000 40 <ENTER> to load 40 additional sectors.
  - 8) Insert your formatted diskette.
  - 9) Type: \$ 30 4000 40 <ENTER> to save these 40 sectors.
  - 10) Your backup diskette is now complete.

- 32K or 48K:
- 1) Format a diskette using your TRSDOS FORMAT program.
  - 2) Load RSM-2D, either version 2 or 3 (32K or 48K).
  - 3) Type: L 0 4000 70 <ENTER> to load 70 sectors into memory.
  - 4) Insert your formatted diskette.
  - 5) Type: \$ 0 4000 70 <ENTER> to save 70 sectors to your diskette.
  - 6) Your backup diskette is now complete.

The first 7 tracks (Sectors 0 to 69) of your diskette contain the RSM-2D monitors and the loader. You have 280 sectors available for your own use, starting with sector 70. RSM-2D can read or write a maximum of 128 sectors in one operation. Be careful not to read disk into memory where RSM-2D is located! You have the following RAM:

VERSION	RAM LOADING ZONE	NO. OF SECTORS
16K	4000 - 6BFF	44
32K	4000 - ABFF	108
48K	4000 - EBFF	172

## USING TAPEDISK TO CREATE RSM-2D COMMAND FILES

You may use TAPEDISK to place RSM-2D on your TRSDOS diskettes. Load RSM-2D from it's diskette. Return to TRSDOS and access TAPEDISK. Use the F command as follows:

```
F RSM16/CMD:0 6C00 7EFF 6E94 <ENTER>      (For 16K Version)
F RSM32/CMD:0 AC00 BEFF AE94 <ENTER>      (For 32K Version)
F RSM48/CMD:0 EC00 FEFF EE94 <ENTER>      (For 48K version)
```

Type exactly as shown. You may use other names, but always use /CMD as the modifier. The drive number (:0) may be changed, but MUST be specified. The start address (6E94, etc.) assures proper initialization of RSM-2D.

**RSM-2D:** This version is furnished on a self-loading diskette. Simply place the RSM diskette into your drive and press reset. Your screen will ask you to select one of the three versions of RSM on the disk. Answer with 1, 2, or 3, corresponding to your memory size of 16K, 32K, or 48K. The correct version will load and execute. Once loaded, you can reboot TRSDOS and make a CMD file from RSM-2D by using the TAPEDISK command. Save the program from the START to the END address in the memory map table.

### GETTING STARTED

When a satisfactory load is achieved, the screen will be cleared, the copyright notice will be displayed, and the screen will respond with:

COMMAND?

This means the monitor is awaiting input from the keyboard. Entries will be written from top to bottom on the screen (RSM-2/2D writes from the bottom of the screen only), and will scroll off the top when the screen is full. You may clear the screen with the CLEAR key. Now type D followed by ENTER. The screen will fill with lines of hex characters, representing memory starting at location 0. Hit the SPACE bar to stop the scroll. Step through several lines with the SPACE bar. Type any other key and the scrolling will continue. Type BREAK and COMMAND? will reappear. Try the same thing but use A instead of D, and memory will be displayed as ASCII characters. Allow the dump to continue until you see addresses in the 0250 range for LEVEL-I or 1630 for LEVEL-II. Stop the scroll and step line by line with the SPACE bar. You are now looking at the BASIC command table! Now read the following detailed description to find out what else RSM can do. Experiment as you go, you can't hurt your computer from the keyboard! At worst, you may have to reload your RSM monitor.

### COMMAND FORMAT

All commands are single character commands, and may have up to three 16-bit hex addresses following the command. Addresses are separated by spaces. The first space is optional. The following commands are all valid commands:

A	No addresses are required.
A 31A0	31A0 is starting address.
D243F 4024	First space is optional.
M 1234 4365 E080	Three fields are maximum.
D 1 10	Leading zeros are optional.
Z 433621543A 8876680	Equivalent to Z 543A 6680.

Several points are illustrated in the above sequence:

- 1) Many commands do not require addresses.
- 2) Leading spaces are optional.
- 3) Leading zeros are optional.
- 4) If more than 4 hex address characters are entered, only the last four are valid.

The following command features are not readily illustrated:

- 5) For most commands, if addresses are not entered, default values will be assumed.
- 6) All commands must be started with a valid letter. If an invalid letter is entered, it will be ignored.
- 7) All addresses are in hex. If a non-hex character is entered in an address field, it will be ignored.
- 8) Spaces are used to separate the address fields. Only one space is allowed; additional spaces are ignored.
- 9) Any command may be aborted by the BREAK key.
- 10) Any character may be erased by a LEFT-ARROW.

PAUSE - Any command that might use a significant amount of time may be interrupted by entering a SPACE. This is usually used in conjunction with commands that dump memory onto the screen. A second SPACE will cause one line of text to be dumped, with a pause again at the end of the line. Typing any character except a space will cause the dump to continue without interruption. The BREAK key will abort any routine and return control to the monitor.

#### COMMAND EXECUTION

All commands are executed by the ENTER key. Commands may be edited by backspacing (LEFT-ARROW) or may be aborted with the BREAK key up until the time ENTER is typed.

#### EXECUTE AND PRINT

Use of the RIGHT-ARROW instead of ENTER to execute commands will cause the output to be directed to your printer as well as the video screen. When the command is finished or is interrupted by BREAK, the printer will be disabled until RIGHT-ARROW is typed again. You may also use the RIGHT-ARROW at the beginning of a command; this will cause the command itself to be printed, as well as the command output. BREAK will disable the printer and abort the command at any time. The printer may either be connected to the Radio Shack Expansion Interface or to our TRS232 Printer Interface. If you are using the TRS232, use the 'Y' command (see Control Commands below) to set the printer baud rate and other parameters.

#### DETAILED COMMAND DESCRIPTION

Commands may be separated into four groups: Control, Tape, Disk, and Memory. They may also be assigned a number corresponding to the number of address fields that are effective. If a command needs more addresses than are input, default values will be assigned. If more addresses are input than required, the first addresses will be used; extras will be ignored. Certain commands have different functions depending upon the number of address fields entered.

## CONTROL COMMANDS

**G - GOTO:** 1 address. Transfers control to specified address. If no address is entered, control is transferred to address 0

**I - INITIALIZE:** 0 addresses (See INPUT command below). Clears screen and initializes all scratchpad locations. This command is accessed automatically after loading RSM-1, and thereafter when I is typed, or if RSM-1 is re-entered using the SYSTEM command (LEVEL-II).

**I - INPUT FROM PORT:** 1 address. If I is typed followed by a hex address, the input port at the specified address will be read, and the hex value displayed on the screen. The stock Radio Shack computer has one input port (cassette) at address FF.

**O - OUTPUT TO PORT:** 2 addresses. The O command will output a specified hex byte to a specified output port. The first address is the output port, and the second is the byte that will be output. If no addresses are typed, RSM-1 will output FF to port 0. If only one address is typed, RSM-1 will output FF to the specified port.

**U - USER COMMAND:** 0-3 addresses. The U command allows the executive routines of RSM to be used in conjunction with custom commands or programs. When U is executed, program control is transferred to the user address listed below, where the user's program should be located. If a normal return is encountered at the end of the user routine, control will revert to RSM. If address fields are entered with the U command, the first address will be found in the D,E registers, the second in H,L and the third in B,C when control transfers to your user program. During RSM-1 initialization, a return instruction (C9 hex) is placed at the user location, thus if U is typed accidentally, no harm is done as control returns to RSM.

RSM VERSION	USER ADDRESS
RSM-1	4880
RSM-1S	4FB0
RSM-2	7F80
RSM-2D (16K)	7F80
RSM-2D (32K)	BF80
RSM-2D (48K)	FF80

**K - KEYBOARD ECHO:** 0 addresses. Allows characters to be typed directly from the keyboard to the screen.

**B - BINARY ARITHMETIC:** 1-2 addresses. This utility routine is useful for converting hex addresses to decimal. If one address is input, the decimal equivalent is output. If two addresses are input, a title line, both addresses, A+B, A-B, and B-A are printed both in hex and in decimal.

**Y - SET TRS232 PARAMETERS:** 0-3 addresses (RSM-2/2D only). TRS232 parameters are set during initialization for a baud rate of 300, 0 null characters, and to output a line feed after each carriage return. To change these values, the 'Y' command is used as follows:

Y B N L <ENTER>

B (1 to 6) selects the baud rate (see table below), N (0-127) is the number of null or fill characters after each carriage return to allow time for carriage travel (see your printer manual), and L enables or disables line feed characters (1=enable, 0=disable). For example, Y 2 5 1 <ENTER> selects 150 baud, 5 nulls and enables line feeds. You are not required to enter all three values if you only want to change the first one or two. Type Y with no numbers to reset the initialization values.

BAUD RATE	CODE NUMBER	BAUD RATE	CODE NUMBER
110	1	450	4
150	2	600	5
300	3	1200	6

**@ - BREAKPOINT AT:** 0-1 address (RSM-2/2D only). @ will set, reset and display software breakpoints for program debugging or tracing. Breakpoints can only be set in read/write memory (RAM). To set a breakpoint, type @ LLLL where LLLL is the hex location where you want the breakpoint. You will see:

BREAK AT LLLL

Entering a second breakpoint will automatically reset the first one. Type @ 0 to reset all breakpoints (a break at 0 is not allowed). Type @ with no addresses to display the current breakpoint (if no breakpoint is set, BREAK AT 0000 will be displayed). For example, type @ 6000 to set a breakpoint at 6000. Now type G 6000 to execute. When the breakpoint is reached, the break is automatically reset, control returns to RSM-2/2D, and all registers are displayed as follows:

BREAK AT 6000

```
AF  BC  DE  HL  IX/Y  STAK
0360 0000 6000 7000 4173 6F7E
0054 1ACE 103D 9231 5033 6DCF
```

(The above values will not be the same on your screen). The hex value of each of the registers at the time the break was reached is displayed. For the AF, BC, DE, and HL registers the first row corresponds to the primary and the second row to the alternate Z-80 register set. The IX and IY registers are in the first and second rows respectively. The first row under STAK is the stack pointer and the second row is the address from the top of the stack. **WARNING:** breakpoints are set by placing a 3-byte call at the break address. Some breakpoint locations can cause your program to "bomb". Avoid locating breaks at RET or JR commands. The 3-byte code will replace part of the code in the following routine, which can cause unpredictable results!

## TAPE COMMANDS

There are four tape commands in the RSM-1/1S monitor: R (read), W (write), L (load and execute) and C (check). For most applications, only R and W will be used. Assume that a user program has been typed into the computer starting at address 4880, and that there are 80 hex bytes in the program, extending to 48FF. To save this program, set up your recorder to record, type W 4880 48FF and ENTER. With LEVEL-I, there is no visible indication on the screen while the tape is writing. LEVEL-II will display a single blinking star. After the tape is written, the checksum will be displayed on the screen.

RSM-1/1S automatically computes the length of the program from the start and end addresses. The length and the start address (in this case 0080 and 4880) are written onto the tape as the first four bytes. To read the tape back into the TRS-80 at the same location, simply set the cassette at the proper location, then type R and ENTER. RSM-1 will read the first four bytes and use them to load and check the tape. The checksum will be displayed at the end of the load procedure. If you desire to load your tape at a different address, type R followed by the desired address. RSM-1 will still read the length from the tape, but will use the typed start address instead of the address on the tape. Again, during tape reading there is no indication with LEVEL-I and a single blinking star with LEVEL-II.

There are two additional tape commands in the RSM-2/2D monitor, R 0 to read SYSTEM tapes, and P to punch or write SYSTEM tapes. The special address 0 MUST be typed after R to read SYSTEM tapes! Any other value will cause RSM to revert to the RSM-1/1S operating mode (see above).

TAPE WRITE COMMANDS

**W - WRITE TAPE:** 2 addresses. Writes cassette tape with header format. Writes checksum at end of tape. Addresses are start and end memory locations. Tape writing may be aborted by use of the BREAK key. After tape writing is complete, the checksum will be written onto the screen. Note: Tapes written by RSM-1 will not load using CLOAD or SYSTEM, but must be read using the RSM-1 R or L commands. With LEVEL-II systems, a single star in the upper right corner of the screen will blink each time 256 bytes are written (about 4 seconds).

**P - PUNCH TAPE:** 2-3 addresses (RSM-2/2D only). Writes LEVEL-II SYSTEM tapes. The first and second addresses are the start and end addresses of the block of memory you wish to place on tape. If you enter a third address, this will be the execution address when you type a / after you load the tape (see SYSTEM command documentation in your BASIC manual). If you do not enter this third address, RSM will use the first address entered as the execution address. After you type ENTER, RSM will ask you:

NAME?

Enter the name you wish to place on the tape. You may use the left-arrow to backspace. Only 6 characters are allowed; RSM will not permit more. ENTER will start the recorder and write your SYSTEM tape.

TAPE READING COMMANDS

**R - READ TAPE:** 0-1 address. Reads RSM tapes into memory. If no address is typed, the start address and length are read from the tape header. If one address is typed, only the length is read from the header; the typed address is used for the start location. Tape errors cause CHECKSUM ERROR followed by the tape checksum and the computed checksum to be displayed. With LEVEL-II systems, a single star in the upper right corner of the screen will blink each time 256 bytes are read.

**R 0 - READ SYSTEM TAPE:** (RSM-2/2D only). This special version of the R command reads LEVEL-II SYSTEM tapes. Prepare your recorder and type R 0 <ENTER>. When loading begins, the tape file name will be displayed. After loading, the execution address on the tape will be displayed. Execute the program with the G command if desired. **WARNING:** the SYSTEM tape will load at its normal operating address. If this conflicts with your RSM monitor, you may not be able to load successfully.

**L - LOAD AND GO:** 0-1 address. Operates the same as R, but program control is transferred to the start address of the loaded program after loading (RSM tapes only). Tape errors will output the CHECKSUM ERROR message and control is not transferred. RSM-2D: see DISK COMMANDS for use of L with 2 or 3 addresses.

**C - CHECK TAPE:** 0 addresses. Operates the same as R without loading the program into memory. Use after writing a tape to verify accuracy. Computes checksum from information read from tape and compares it with checksum recorded on tape.

ABORTING THE TAPE COMMANDS

If you inadvertently type W or wish to discontinue a tape writing activity, simply type BREAK. This will return control to RSM without completing the tape.

Escape from the R, L, or C commands in RSM-1/1S requires an additional step. BASIC tape reading routines are used, which have no escape provisions. If you type R without a tape loaded, the computer hangs up in BASIC waiting for a tape signal. To escape without having to reload RSM-1/1S, simply start any recorded tape running and press BREAK. The tape signal causes a return from BASIC to RSM-1/1S where BREAK detection software can abort the command. RSM-2/2D contains its own tape reading subroutines with built-in BREAK detection, thus it is only necessary to type BREAK to abort tape reading functions.

**DISK COMMANDS (RSM-2/2D ONLY)**

**L - LOAD FROM DISK:** 2-3 addresses. Reads directly from specified disk sector into specified memory location. First address is disk sector in decimal (0-349). Second address is memory location in hex. Third address is number of disk sectors to load, also in decimal. If only two addresses are entered, only 1 sector will be loaded.

**\$ - SAVE TO DISK:** 2-3 addresses. Writes directly from memory to specified disk sectors. First address is disk sector in decimal (0-349), second address is memory start location, and third address is decimal number of disk sectors to write. Again if only two addresses are entered, only 1 sector will be written.

## MEMORY COMMANDS

Memory may be displayed, modified, moved, exchanged, or tested. All memory display commands may be stepped through one line at a time by using the SPACE bar, or terminated by use of BREAK. Unless otherwise stated, memory commands are addressed as follows: the first and last addresses are the starting and ending locations respectively. If no addresses are input, dump will start at 0 and continue until aborted. Entering one address will cause a continuous dump starting at the typed address. Two addresses will dump memory from the first address through the second address. SPACE will stop dump at the end of the current line; additional spaces will output single lines. Any other character (except BREAK) will cause the continuous dump to resume.

**A - ASCII DUMP:** 0-2 addresses. Interprets and displays memory as ASCII characters. Non-ASCII characters and control characters are displayed as periods. Each line starts with the address of the first location displayed, followed by 48 characters, unless a carriage return is encountered, which will cause a new line to be started. This feature formats ASCII files for easy reading.

**A SSSS 0 - FORMATTED ASCII DUMP.** If two addresses are input and the second is 0, a formatted ASCII dump is output. This command lists memory with 16 characters per line, starting at address SSSS. Printable ASCII characters are listed normally. If the 8th bit of a printable character is set, the character will be followed by a period. If the character is non-printing, the hex value of the character will be printed.

**D - HEX DUMP:** 0-2 addresses. Interprets and displays memory as two hex characters per byte. Start address is displayed at the beginning of each line; 16 bytes are displayed per line.

**E - EDIT MEMORY:** 1 address. Displays address and hex value of input address followed by a dash (-). This location may be modified by entering two hex characters. If more than two characters are entered, the last two will be used (the BACK-ARROW is not active for this command), thus errors may be corrected by retyping the correct values. Entering a space will transfer the screen entry to memory and display the following memory location. If nothing is typed after the dash, the spacebar will step to the next location without changing the memory location. The ENTER key will have the same effect, but a new line and new address will be displayed on the screen. Typing UP-ARROW will back up one byte, and a new line with the new address will be output. Exit the routine with BREAK, which will also transfer the last displayed byte into memory.

**F - FIND 1 BYTE:** 3 addresses. Searches through memory block defined by the first two addresses and finds all locations with the byte specified in the third address field. Writes address, the located byte, and the following byte (in hex) on the screen. This command is designed to locate input/output commands with their arguments. It can also be used for other searches, of course, with the proper interpretation of the output arguments.

**H - HUNT 2 BYTES:** 3 addresses. Searches memory from first to second address for two-byte combinations specified by third address. Address is interpreted in 'reverse' notation (low byte first). Address of location preceding the found address is output, followed by the code and finally followed by the address. This command is designed to locate references to memory locations and to identify the referencing command, but may be used for other searches, provided 'reverse' entry is used for the desired two-byte combination.

**M - MOVE MEMORY:** 3 addresses. Moves a block of memory between the first and second addresses to a similar block starting at the third address. There are no restrictions on addressing - overlap from either end is allowed.

**Q - CHECKSUM:** 2 addresses. Computes and displays checksum for memory block between two addresses. Used to determine if program has been 'bombed' during operation.

**S - SYMBOLIC DUMP:** 0-2 addresses. Displays memory in standard Zilog mnemonics. All absolute and relative addresses (jumps, calls, etc.) are computed and output in hex code. This command is not included in RSM-1.

**T - TEST MEMORY:** 2 addresses. Uses random numbers to test memory between two specified addresses. Writes address, expected value and actual value for locations where errors occur. Writes number of tests completed as they are finished; 256 checks of each location is defined as 1 test.

**V - VERIFY MEMORY:** 3 addresses. Compares block of memory from first address to second address with block starting at third address. Writes address and both values where differences occur.

**X - EXCHANGE MEMORY:** 3 addresses. Exchanges block of memory between first and second addresses with block starting at third address.

**Z - ZERO MEMORY:** 2-3 addresses. Writes code in third address into block of memory between first two addresses. For two addresses, 00 is written into the block between the addresses. As with all commands, addresses are inclusive.

### ADDITIONAL FEATURES

#### JUMP TABLE

RSM-1/1S has three jumps located at the beginning of the program at addresses 4203, 4206, and 4209. The first jump is to the starting point and is used for re-entering RSM-1/1S. The second jump at 4206 is to the CRT output routine. You may use this location to write directly to the screen, with normal scrolling, etc. The character to be output should be in the A register. The third jump at 4209 is to the keyboard routine. If you call this location, the keyboard routine will not return until a key has been pressed. The typed character will be returned in the A register.

RSM-2/2D has 5 jumps located at the beginning. The first three, starting at 6C00 (or BC00/EC00 for RSM-2D) are to the monitor start, CRT, and keyboard, as above. The 4th jump is to a BREAK check; it will return from a call with no effect unless the BREAK key has been pressed. BREAK will return control to RSM-2. The 5th jump is to the PRINT driver. The character in the A register will be output to the printer.

## MACHINE LANGUAGE PROGRAMMING

Machine language programming is a skill that must be learned by a combination of study and practice. A minimum requirement for success is the ZILOG Z80 technical manual and a good book on programming with the Z80. One that we have found helpful is "The Z-80 Microcomputer Handbook" by William Barden Jr., from Howard W. Sams & Co., Inc., for \$8.95.

The following is an example of a machine language program for placing a full set of 8 bit characters on your TRS-80 screen:

ADDR	CODE	LABEL	MNEMONIC	COMMENT
4FD0	21 00 3C		LD HL,3C00H	;POINT TO SCREEN
4FD3	36 20	CLEAR:	LD (HL),20H	;STORE A BLANK
4FD5	23		INC HL	;ADVANCE POINTER
4FD6	CB 74		BIT 6,H	;CHECK FOR H=40
4FD8	28 F9		JR Z,CLEAR	;DO ENTIRE SCREEN
4FD5	21 00 3D		LD HL,3D00H	;POINT TO LINE 5
4FD8	75	LOOP:	LD (HL),L	;PUT CHAR. ON SCREEN
4FD9	2C		INC L	;ADVANCE POINTER & CHAR.
4FDA	20 FC		JR NZ,LOOP	;DO IT 256 TIMES
4FDC	C9		RET	;RETURN TO RSM

Enter the program by typing E 4FD0 then ENTER. You will see:

4FD0: 00-

(The hex code may be 00 or any other code). Type 21 followed by a space (or ENTER), then 00 and another space, etc., until you have all of the code entered. Check the code with the D command or the S command if you have RSM-1S. A hex dump of the program should appear as follows:

```
4FD0: 21 00 3C 36 20 23 CB 74 28 F9 21 00 3D 75 2C 20
4FE0: FC C9 XX XX
```

where XX can be any hex code. Now type G 4FD0 and ENTER. If you made no mistakes, your program will run.

This program will clear the screen, then place 4 rows of characters on the screen, starting about the 4th line from the top. You will find there are 64 ASCII and 64 graphic characters, both repeated twice. Your screen is a block of 1024 bytes of memory located from 3C00 to 3FFF. However, this memory only has 7 bits - bit 6 is missing! This is the reason the characters repeat twice. The most significant bit (bit 7) determines whether the display is graphics or ASCII.

## IN CASE OF TAPE DIFFICULTY

If your cassette fails to load, take the following steps:

- 1) Be sure the tape is recorded. If you can't hear your tape while it's loading, pull the plug from the monitor jack and listen. Note the counter setting where the program begins and ends. Write down the numbers for BOTH copies on the tape. Consider placing a resistor (100-470 ohms) across the speaker disable switch inside the tape recorder so you can hear your tapes (at reduced volume) while they are loading.
- 2) Be sure your tone control is at its highest setting. Experiment with the volume control. Nearly all the tapes that have been returned to us will load at some volume setting. LEVEL-II is particularly touchy.
- 3) Be sure you leave the tape on long enough; check your counter. For LEVEL-I the stars on your screen are not a good indicator; they only blink when an "ENTER" character is detected. For machine language programs the ENTER character is random, thus the blinking star is unpredictable. The LEVEL-II star blinks each 256 bytes (about 4 seconds).
- 4) Try both copies on the tape at several volume settings. These procedures will solve nearly any tape loading problem. However, if you are still unsuccessful, return your cassette for a replacement.

## SOURCE LISTING

Many RSM routines are useful for your own machine language programs. You may purchase a source listing for RSM-1, either LEVEL-I or LEVEL-II for \$5.00, or both for \$7.50 if ordered together. LEVEL-I is in INTEL 8080 mnemonics and LEVEL-II is in ZILOG Z-80. The memory map is identical, but the code is not.

RSM-2/2D owners may purchase the source for the 6C00 version of RSM-2 (ZILOG mnemonics) for \$7.50. RSM-2 includes tape and disk reading and writing routines, a CRT driver, print software and a keyboard scanner.

Please note that the source listings contain the monitor only; the source for the symbolic dump routine is not included. You may order the source when you send in the registration form, or you may order the source at a later date. However, only registered owners may purchase the source listings. Registration will also place you on our mailing list.

If you own RSM-1, we offer RSM-1SU for \$9.95 postpaid. RSM-1SU loads into the user area of RSM-1 and provides the symbolic dump command for RSM-1 owners.

=====  
Copyright (C) 1978 by Small System Software, all rights reserved. Small System Software makes no representations or warranties with respect to the contents of this documentation and tape cassette or diskette, and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Small System Software is not liable for consequential commercial damages.

COMMAND SUMMARY FOR RSM MONITORS

RSM-1S commands are listed below. Several commands have dual functions depending upon the number of address fields following the command letter. RSM-1 commands are identical to RSM-1S except the Symbolic dump (S) is not included. Page numbers refer to command descriptions in this documentation.

COMMAND	PAGE
A - ASCII DUMP: Display ASCII equivalent of memory block	8
B - BINARY ARITHMETIC: Add/subtract in hex and decimal	4
C - CHECK TAPE: Check cassette tapes for proper checksum	7
D - HEX DUMP: Display Hex equivalent of memory block	8
E - EDIT: Enter, examine, or modify memory in hex code	8
F - FIND 1 BYTE: Find and display single-byte hex codes	8
G - GOTO: Go to and execute program at specified address	4
H - HUNT 2 BYTES: Find and display two-byte addresses	9
I - INITIALIZE/INPUT: Initialize or input data from port	4
K - KEYBOARD ECHO: Type directly to screen or terminal	4
L - LOAD AND GO: Load cassette tape and execute program	7
M - MOVE: Move any block of memory to specified location	9
O - OUTPUT: Output hex value to specified output port	4
Q - CHECKSUM: Compute checksum of specified memory block	9
R - READ TAPE: Read cassette tape (header or standard)	7
S - SYMBOLIC DUMP: Display memory in ZILOG Z-80 mnemonics	9
T - TEST MEMORY: Test memory block and display errors	9
U - USER: Allows user to write and execute new commands	4
V - VERIFY MEMORY: Compare any two blocks of memory	9
W - WRITE TAPE: Write any memory block to cassette tape	6
X - EXCHANGE: Interchange any two blocks of memory	9
Z - ZERO MEMORY: Write zero or any hex code into memory	9

RSM-2/2D includes all of the above RSM-1S commands plus the following additional commands (the R and L commands become dual function commands):

@ - BREAKPOINT: Inserts breakpoint AT specified address	5
P - PUNCH: Writes cassette tape in LEVEL-II SYSTEM format	6
R - READ: Read SYSTEM tape, display name, start address	7
L - LOAD: Loads specified disk sectors into memory block	7
\$ - SAVE: Saves memory block into specified disk sectors	7
Y - TRS232: set print parameters (baud, etc.) for TRS232	5
-> - PRINT: Directs output to printer as well as screen	3

**SMALL  
SOFTWARE  
SYSTEM**

SMALL SYSTEM SOFTWARE  
Post Office Box 366  
Newbury Park, CA 91360

**SMALL  
SOFTWARE  
SYSTEM**